A progressive filtering heuristic for the location-routing problem and variants

A progressive filtering heuristic for the location–routing problem and variants

Florian Arnold[a], Kenneth Sörensen [1a]

[a] *University of Antwerp, Departement of Engineering Management, ANT/OR - Operations Research Group*
[b] *corresponding author. Email: florian.arnold@uantwerpen.be*

Abstract

The location–routing problem (LRP) unites two important challenges in the design of distribution systems: planning the delivery of goods to customers (i.e., the *routing* of the delivery vehicles) and determining the locations of the depots from where these deliveries are executed.

In this paper, we design an efficient and effective heuristic for the LRP based on an existing heuristic to solve the capacitated vehicle routing problem. Our heuristic reduces the solution space to a manageable size by the estimation of an upper bound for the number of open depots and then iteratively applies the routing heuristic on each remaining depot configuration. A progressive filtering framework, in which the vehicle routing problem is solved to a larger precision at each iteration, is employed to quickly detect unpromising configurations.

Extensive experimentation reveals that the estimated upper bound effectively reduces the search space on different types of instances and that a good filtering design combines coarse and fine filters. Benchmarking shows that, despite its simple design, the final heuristic outperforms existing heuristics on the largest LRP benchmark set, on very-large-scale LRPs, and on 2-echelon LRPs.

Keywords: vehicle routing problem, heuristics, location routing problem, large-scale problem, local search

1. Introduction

The location–routing problem (LRP) is a well-known combinatorial optimization problem that combines two important supply chain decisions: where to open distribution facilities or depots (*location*) and how to organize the distribution of goods from those depots to customers (*routing*). In practice, both decisions are often taken on different time horizons: while the opening of facilities is a long-term, *strategic* decision, vehicle routes are planned on an *operational*, day-by-day basis.

[1] corresponding author. Email: kenneth.sorensen@uantwerpen.be

Despite the different planning horizons, however, the LRP is not devoid of practical applications. Salhi and Rand (1989) have shown that the incorporation of distribution planning into a facility location decision can significantly improve the overall quality of the solution to the location subproblem, resulting in a considerable decrease in the total supply chain cost.

Moreover, in some situations, the planning horizons do not differ that much, if at all. Schittekat and Sörensen (2009), e.g., describe an LRP in which the facilities are owned by third-party logistics (3PL) providers, which means that the selection of facilities can change in a manner of weeks, and routes can remain the same for several weeks. Additionally, in cases where "depots" are inexpensive and/or mobile, the location decisions are made, like the routing decision, on an operational level. An example can be found in the context of waste collection (Del Pia and Filippi 2006) where smaller vehicles need to synchronize with larger ones to dump their collected waste. As argued by Vidal et al. (2020), LRPs might also be extended with multiple routing scenarios to achieve a more realistic representation of many tactical location routing models.

Informally, the LRP can be summarized as follows. Given a set of customers with known demand and a set of candidate facility locations (we will use the term "depot" in the remainder of this paper) with a fixed opening cost, determine which depots to open as well as the delivery routes from the open depots to the customers to minimize the sum of depot opening costs and routing costs. Travel costs between customer pairs and between customers and depots, as well as fixed vehicle costs, are known, and each customer should be visited once on a route starting and terminating at an open depot. The total demand in each route cannot exceed the (homogenous) capacity of the vehicles.

The LRP is a generalization of the capacitated vehicle routing problem (VRP) and more specifically the *multi-depot* vehicle routing problem (MDVRP). While the set of depots is fixed in MDVRPs, in LRPs a subset of depots can be selected from a set of candidate depots, thus adding elements of the facility location problem (FLP) and increasing complexity. After a subset of depots has been selected (we will refer to this subset as *depot configuration* in the remainder of this paper), the LRP reduces to an MDVRP.

A straightforward idea to solve an LRP is to enumerate all possible depot configurations, to compute a routing solution for each of them, and finally to select the best configuration together with the solution of the respective MDVRP as the solution of the LRP. In this context, we call the computation of a routing solution the *evaluation* of the respective depot configuration. The obvious bottleneck of this approach is the computation time required, equal to the time to evaluate a single depot configuration multiplied by the (exponential) number of depot configurations.

In this paper, we present two key ideas to circumvent this complexity and examine all promising depot configurations. These ideas are embedded in an algorithm coined *progressive filtering* (PF). First, and before any routing solution is calculated, the algorithm eliminates all depot configurations with more depots than a certain upper bound.

This upper bound is derived as a function of the properties of the particular instance. For smaller problem instances, this restriction of the set of depot configurations suffices, and it becomes tractable to enumerate them all. For larger instances, in which the number of depot configurations is still large after setting a bound on the number of open depots, the algorithm uses a heuristic procedure to limit the number of depot configurations to examine. In the second step, the algorithm progressively filters the remaining depot configurations using an increasingly accurate version of the MDVRP heuristic knowledge-guided local search (KGLS, Arnold and Sörensen (2019)). In this way, the worst depot configurations are eliminated using a low-accuracy, fast routing algorithm, leaving more computing time to evaluate the better depot configurations using a more accurate (and therefore slower) routing algorithm.

We demonstrate that the resulting algorithm is competitive with the most effective LRP heuristics in literature by applying it to a wide range of benchmark instances. Moreover, the heuristic can be readily adjusted to solve very-large-scale LRP instances with thousands of customers and hundreds of depots, the 2-echelon-LRP (2E-LRP, see, e.g., Drexl and Schneider (2015)) and the single truck and trailer routing problem with satellite depots (STTRPSD, see, e.g., Villegas et al. (2010)). On all problem classes, the heuristic either outperforms or matches the performance of all previously published heuristics in the literature. In summary, this work makes the following contributions to the field of heuristics and location–routing problems.

- The introduction of a flexible and extensible heuristic framework for hierarchical decomposable problems called progressive filtering.
- An empirical estimate for an upper bound of open depots in LRPs and a heuristic construction procedure that both significantly limit the number of promising configurations.
- A heuristic for LRPs and LRP variants that matches or improves the performance of state-of-the-art heuristics on several problem sets.

The remainder of this paper is organized as follows. In Section 2 and Section 3 we present effective heuristics and ideas from the literature and formally introduce the location–routing problem and its variants. Section 4 introduces the progressive filtering framework together with the MDVRP heuristic KGLS. Section 5 investigates how the number of depot configurations examined by the heuristic can be restricted. Different setups of the filtering framework are thoroughly analyzed in Section 6 to arrive at a final setup which is compared with effective heuristics on various benchmark sets. We conclude with a summary of our findings as well as an outlook for future research in Section 7.

2. Literature Review

In recent years, several effective heuristics have been proposed to solve the LRP. Albareda-Sambola and Rodríguez-Pereira (2019) and Schneider and Drexl (2017) give a

concise overview about the recent progress on heuristics for the LRP. Prodhon and Prins (2014) also review heuristics for LRP variants. The most successful heuristics identified by Schneider and Drexl (2017) consist of different stages and combine (meta)heuristic components with integer linear program formulations (ILPs).

Escobar and Linfati (2013) and Escobar et al. (2014) successfully use ILP to determine a depot configuration. Initially, the customers are divided into clusters and an ILP model assigns customer clusters to depots such that the routing cost from the open depots is minimized. Afterward, the routing cost is improved with various heuristic techniques.

Contardo et al. (2014) suggest a heuristic design which solves the location and routing subproblems simultaneously. A pool of candidate solutions is constructed with a randomized version of the extended algorithm by Clarke and Wright (1964), which are subsequently improved by iteratively removing and re-inserting customers (destroy-and-repair) and improving the assignment between routes and depots by solving an ILP. Notably, the depot configuration is also improved with local search moves that can open and close depots.

This idea is also incorporated in the state-of-the-art heuristic by Schneider and Löffler (2017). The authors utilize a wide range of granular local search moves to improve the solution for the routing subproblem of an initially generated solution. Alternative depot configurations are then investigated in a tree-like fashion by either swapping an open and a closed depot, closing an open depot or opening one extra depot. The created depot configurations are then evaluated by improving the routing solution with the granular local search.

Despite the success of heuristics in the domain of LRPs, little attention has been devoted to a systematical identification of promising depot configurations. Prins et al. (2006a) assign customers to their closest depot with sufficient capacity and close the depots with no assigned customers. In this way, several depots are eliminated from the subsequent search. Chan and Baker (2005) determine a lower and an upper bound of open depots and open a random number of depots between those bounds. The bounds are computed by opening depots near the minimal spanning tree (MST) formed by all customer nodes. An MST is also used by Harks et al. (2013) to approximate a solution for very-large-scale LRP instances. In comparison, the ant colony optimization of Ting and Chen (2013) initially opens a random number of depots, where depots with a high ratio between depot capacity and opening cost are favored.

From this overview, we can conclude that the MDVRP subproblem is generally solved with heuristics, whereas the location subproblem is more often solved by (exact) integer programming methods, although local search approaches are also used. Both approaches have been shown to work well on instances of smaller and moderate size in which the depots are relatively homogeneous in terms of capacity and costs, and few depots have to be opened.

However, ILP formulations and exact methods might face difficulties if the problem grows beyond a certain size and the location subproblem becomes the computational bottleneck. On the other hand, local search based approaches might struggle on in-

stances in which promising depot configurations are diverse (e.g., there are good solutions with a single open depot and with 20 open depots). While exact methods can handle such instances with structurally different local optima because — by their very nature — they examine the entire solution space, such instances present difficulties for local search, as improving paths between high-quality solutions do not exist.

Drawing on this insight, we develop an alternative framework to solve the LRP that attempts to examine as many potentially optimal depot configurations as possible while making sure to discard unpromising depot configurations as early as possible. To do this intelligently, we first investigate the properties of promising depot configurations to initially reduce the search space as much as possible.

3. Problem definition

The LRP combines the problem of determining a set of open depots from a set of candidate depots, with the problem of routing a fleet of vehicles from the open depots to the set of customers. Let G be a complete, weighted, and undirected graph with nodes $V = I \cup J$. Each edge $(i, j) \in V \times V$ of G is annotated with a cost $c(i, j)$. Nodes $J = \{1, 2, \ldots, n\}$ represent customers, while nodes $I = \{n+1, n+2, \ldots, n+m\}$ represent candidate depots. Each depot $i \in I$ has a fixed opening cost O_i and a capacity W_i. Among all candidate depots, a *depot configuration*, i.e., a subset of open depots $D \subseteq I$ needs to be determined.

From the set of open depots D, each customer $j \in J$ with demand d_j has to be visited on one delivery route. The delivery routes R are planned with a homogeneous fleet of vehicles where each vehicle has a maximum capacity Q and a fixed cost F. Overall, the following constraints have to be considered when planning the delivery routes: (1) each customer is visited exactly once, (2) each route returns to the same depot it started from, (3) the sum of demand of all customers on a route does not exceed the vehicle capacity Q, and (4) the sum of demand of all customers allocated to routes of depot i does not exceed this depot's capacity W_i. A distinction can be made between *capacitated* instances, in which some depots are not able to serve all demand ($\exists i | W_i < \sum_j d_j$), and *uncapacitated* instances, in which the capacity of each depot is assumed to be infinite.

The objective is to find a depot configuration $D \subseteq I$ and a corresponding routing solution with minimal total cost $c_{\text{LRP}}(D) = c_O(D) + c_R(D)$. The *opening costs* of a specific depot configuration are given by $c_O(D) = \sum_{i \in D} O_i$ and its *routing costs* can be calculated as $c_R(D) = F \cdot \sum_{(i,j) \in R | i \in I} 1 + \sum_{(i,j) \in R} c_{ij}$.

In the last decade, many variants to this standard version have been introduced to tackle a variety of real-world problems (Drexl and Schneider 2015). A taxomy of these problems is presented by Lopes et al. (2013). In this paper, we consider the 2-echelon LRP (2E-LRP) and the single truck and trailer routing problem with satellite depots (STTRPSD) as variants.

The 2E-LRP is a generalization of the LRP with a two-level distribution structure (see, e.g., Drexl and Schneider (2015)). On the first level, the transportation from

a single main depot (also called *platform*) to various subsidiary depots (also called *satellites*) has to be planned, while on the second level, the subsequent distribution from satellites to customers is planned. The satellites have opening costs and a subset among all possible satellites has to be determined such that the sum of opening costs, routing costs on the first level, and routing costs on the second level is minimized. Once a subset of satellites has been determined, the first level presents a VRP, and the second level presents an MDVRP. More formally and analogously to the definition of an LRP, the set of candidate satellites of the 2E-LRP corresponds to the set of candidate depots I, a depot configuration is defined by $D \subset I$ and the set of customers by J. Let P denote the platform, R_1 the routing solution on the first level and F_1 the fixed cost per route on the first level, then the costs of a configuration D on the first level are expressed as $c_F(D) = F_1 \cdot \sum_{(i,P)\in R_1} 1 + \sum_{(i,j)\in R_1} c_{ij}$ and the overall costs as $c_{\text{LRP}}(D) + c_F(D)$. Note that the vehicle capacity on the first level can differ from that of the second level (it is usually assumed to be higher). Similarly, the fixed cost per route is usually higher on the first level.

The STTRPSD constitutes a special case of the 2E-LRP in which satellite depots have opening costs $O_i = 0$ and infinite capacity W_i (Villegas et al. 2010). A set of customers is delivered from a platform by a single truck with an attached trailer. As an additional constraint, the trailer cannot be attached to the truck when visiting a customer (e.g., because there is a lack of parking space). Thus, before visiting a subset of customers, the truck has to be detached and parked at one of several possible parking locations (without extra costs). After the subset of customers has been delivered, the truck returns to the parking location to re-attach the trailer.

4. Progressive filtering

In this paper, we propose a heuristic for the LRP which we call *progressive filtering*. PF combines existing heuristics for the MDVRP with an iterative filtering strategy to discard unpromising depot configurations as early in the search as possible.

4.1. Efficiently solving capacitated MDVRPs

PF uses two different heuristics to compute routing solutions for MDVRPs. The first heuristic (Regret Clarke-Wright - RCW) is a simple construction heuristic that allocates each customer to its closest open depot and solves the corresponding VRP with the heuristic by Clarke and Wright (1964) for each depot separately. The second heuristic is a slightly modified configuration of the Knowledge-Guided Local Search (KGLS) heuristic that we have proposed in Arnold and Sörensen (2019). Both heuristics were adapted to deal with capacitated depots in LRP instances.

RCW allocates each customer to its nearest depot with sufficient capacity. The order of this allocation is determined by minimizing the opportunity costs that are incurred when the closest depot does not have sufficient capacity to serve the customer (i.e., the regret). For each customer, the difference between the distance to the closest and the second-closest depot is computed, and the customer with the largest difference

is allocated to its closest depot first. If the closest depot has insufficient capacity left to serve the considered customer, the regret is updated by computing the difference between the distance to the two nearest depots with sufficient capacity. After each customer has been allocated to a depot, the initial routing solution is constructed by applying the Clarke and Wright heuristic to each depot separately. The result is either used as a routing solution if computation time is limited, or serves as an initial solution for KGLS to achieve further improvement.

KGLS is a local search based and deterministic metaheuristic with a single solution trajectory that combines the guided-local search framework (Voudouris and Tsang 2003) with sophisticated and complementary local search operators. Multiple routes are improved with the CROSS-exchange operator and a relocation chain. Whenever an improving local search move has been found, the involved routes are re-optimized with the heuristic by Lin and Kernighan (1973). When no more improving moves can be found and a local minimum is reached, a subset of undesirable edges is identified with problem-knowledge from a preceding data-mining study. These edges are penalized and the heuristic then attempts to remove them with the previously mentioned local search operators. The local search uses pruning techniques and sequential search to find improving moves as fast as possible and to scale effectively to very-large-scale instances (Arnold et al. 2019). As a consequence, KGLS consistently produces high-quality solutions for routing problems in short computation times. On MDVRP instances with 50 to 360 customers, KGLS requires less than a minute of computation time to find solutions with an average gap of 0.08% to the best-known solutions (Arnold and Sörensen 2019).

The capacity constraint on the depots is embedded by keeping track of the remaining inventory at each depot and only allowing local search moves that do not violate these constraints. Note that these constraint checks are only relevant if a local search move attempts to exchange customers from routes that are assigned to different depots. As a result, the heuristic keeps solutions feasible at all times. Algorithm 1 provides a high-level outline of the KGLS heuristic and its adaption to MDVRPs with capacitated depots.

Algorithm 1 Knowlege-guided local search heuristic (KGLS), adapted for MDVRPs with capacitated depots.

1: Construct an initial solution S with RCW.
2: Apply Local Search on S with capacity checks
3: *iter = 1*
4: **while** *iter*$< I_{\text{MAX}}$ **do**
5: Penalize undesirable edges in S until 30 moves have been made
6: Apply Local Search on S with capacity checks
7: *iter++*
8: **end while**

4.2. Solving LRPs through progressive filtering

Even with a fast routing heuristic like KGLS, calculating an accurate solution for a moderately-sized MDVRP requires at least some seconds of computation time. This time magnitude might be too large if there are many candidate depots (an LRP with m candidate depots has $2^m - 1$ depot configurations). This dilemma can be solved by evaluating each configuration only as long as necessary.

Many configurations can be ignored by the search without even computing any routing solutions. More precisely, we hypothesize that only a limited number of depots have to be opened, beyond which the sum of opening costs exceed any potential benefit in terms of decreasing routing costs. The estimation of such an upper bound U would sharply reduce the number of promising depot configurations to $\sum_{i=1}^{U} \binom{m}{i}$. However, U or m (or both) might still be too large. In the case of the STTDR, such a value cannot be estimated because depots can be opened for free. In these cases, the number of remaining configurations is still larger than an acceptable limit M and further elimination is necessary. We suggest an elimination based on a divide and conquer approach, which takes the individual qualities of each depot into account. These ideas are elaborated in Section 5.

The evaluation of the remaining configurations requires the computation of routing solutions. However, the computation should only be carried out as long as necessary. The key idea is that a depot configuration D_1 is only evaluated until there is *sufficient confidence* that there is a better configuration D_2. If there is sufficient confidence that such a better configuration exists, we can stop the evaluation and remove D_1 from the search. The level of confidence depends on the difference in solution quality to the best configuration so far (the larger the gap between the quality of both configurations, the more confidence) and the expected accuracy of the solution method (the higher the expected accuracy, the more confidence). Even with a very fast but rather inaccurate evaluation of the routing solutions for different depot configurations (e.g., with RCW), some configurations will be clearly outperformed by others. These configurations can be "filtered" out at this stage. In a next stage, a slightly more accurate heuristic can be used, which takes more time, but can confidently filter out more configurations. This process can be repeated until just a few configurations remain, which are then evaluated with the most powerful routing heuristic available. The overall goal is to iteratively reduce the number of configurations, while not filtering out the best or one of the best configurations. We call this process *progressive filtering*.

In this context, a filter F is defined as a selection process, which selects a subset of size b of output configurations C_O from all given input configurations C_I, using algorithm a as evaluation criterium. This is formalized as $F_{a,b}(C_I) = C_O$ with $|C_O| = min(b, |C_I|)$. In the following, the b configurations with the lowest objective value are selected, and algorithms RCW and KGLS are used as evaluation algorithm a. A filter is thus a tuple formed by the used algorithm a and its abortion criterium and the maximal number b of configurations that are passed on to the next filter (also called granularity in the following). Then a set of filters $P = \{(a_1, b_1), (a_2, b_2), \dots\}$ defines a setup for PF. The entire framework is outlined in Algorithm 2.

In summary, PF is a flexible algorithmic framework for the LRP and other problems that are decomposable into two or more hierarchical subproblems. Algorithm a can be treated as a black box that solves the subproblem with given accuracy and returns a problem-specific objective value. PF can thus also be readily extended to other LRP problem variants.

Algorithm 2 The progressive filtering framework (PF).

Input: Set C of all possible depot configurations

1: Estimate an upper bound for the number of open depots U and remove all configurations D with $|D| > U$ from C
2: **if** $|C| > M$ **then**
3: Heuristically select promising configurations from C
4: **end if**
5: **for** $(a, b) \in P$ **do**
6: $C \leftarrow F_{a,b}(C)$
7: **end for**
8: Select the best configuration C_{best} from C and compute a routing solution with a.

4.3. Multi-Threading

PF can be readily parallelized by distributing the evaluation effort at each filtering stage between different threads. This can lead to significant performance gains when larger servers or cloud computing are available, and makes the heuristic framework appealing from a practical perspective when times matters and computational resources are unrestricted.

Let PF_t be an implementation of PF that uses t worker threads for the evaluation of configurations, and one master thread that organizes the distribution of the configurations and collection of results. At each filtering stage i, each worker thread receives $\lceil \frac{b_{i-1}}{t} \rceil$ depot configurations for evaluation. After all threads have finished their evaluation, the results are collected by the master thread, and the best b_i configurations are re-distributed among the worker threads. If $t \geq b_i$ for some stage i, all remaining filtering stages in PF_t can be skipped and the final stage (line 8 in Algorithm 2) can be executed immediately with all b_i remaining configurations.

Note that the maximal performance gain with multi-threading is limited by the run-time of the final stage. KGLS in its current form is single-threaded and, thus, the time required for a single run with a maximum number of iterations cannot be reduced by using multiple threads. The runtime of all other stages decreases proportionally to t in the best case, however, usually the runtime of different threads usually varies since the evaluation time of each configurations varies.

4.4. Solving LRP variants and very-large-scale LRPs

With minor adaptations, PF can solve 2E-LRPs and STTRPSDs. It can also solve LRPs on very-large-scale instances.

2E-LRP. Given a depot (satellite) configuration D, the routing cost on the first level $c_F(D)$ can be computed by, e.g., applying the CW heuristic on the respective VRP. Since the number of candidate satellites is generally small (and magnitudes smaller than the number of customers on the second level), such a simple evaluation should already be sufficient to obtain a good estimate for $c_F(D)$. Thus, the only adaptation necessary is the computation of routing solutions for the first level with CW to obtain $c_F(D)$ whenever a routing evaluation is performed. With this design, solving an 2E-LRP does not require much more computational effort and no additional implementation effort in comparison to solving the corresponding LRP that arises from the decisions which satellite to open and how to distribute goods from the satellites to the customers. We want to remark that this decomposition simplifies the problem and does not consider the interdependency between the first and the second level. The allocation of customers to satellites on the second level determines the demand of the satellites and, thus, the routing solution on the second level determines the routing problem on the first level and vice versa.

STTRPSD. The STTRPSD can be solved analogously to a 2E-LRP. Since opening costs are nonexistent, no upper bound for the number of open depots U can be derived, which makes STTRPSD a challenging problem variant to validate the effectiveness of the heuristic construction of depot configurations.

Very-large-scale LRPs. The progressive filtering framework implicitly relies on the two assumptions that the number of depot configurations can be reduced effectively and that the associated MDVRPs can be evaluated efficiently. For LRPs in which the number of candidate depots or the number of customers is large, e.g, $m \geq 100$ or $n \geq 1000$ (or both), these assumptions no longer hold, and the initial number of depot configurations, as well as the time spent on the subsequent routing evaluation, have to be limited even more drastically to be computationally tractable. We propose three minor adaptations to solve such instances. First, the evaluation of MDVRPs with $n \geq 1000$ is costly and only a small number of configurations can be evaluated in a reasonable amount of time. Thus, each of these configurations is evaluated with the fast version of CW described in Arnold et al. (2019), called CW^{100}, in which the savings list only contains the savings between a customer and its 100 nearest neighbors. In case that the number of depots is not overly large but the number of customers exceeds a certain threshold, i.e., $m < 100$ and $n \geq 400$, CW^{100} is also utilized to speed up the construction process. Secondly, the area division of the heuristic construction outlined in Section 4.2 is simplified by only opening the depot with the lowest gross costs in each area. This simplification limits the number of considered configurations to U. Finally, no filters are used ($P = \{\}$) and the most promising configuration determined by heuristic construction is immediately evaluated with KGLS.

5. Identifying promising depot configurations

In the following, we present two ideas to drastically reduce the number of depot configurations that appear promising. First, we experimentally derive an approximate

upper bound U for the number of open depots in order to limit the search space to $\sum_{i=1}^{U} \binom{m}{i}$ configurations. Second, for those instances in which the number of remaining depot configurations is still too large, we present a simple heuristic procedure to further reduce the number of options.

5.1. Estimation of an upper bound for the number of open depots

While the opening of depots usually comes at a cost, it can be expected that more open depots allow for a more efficient routing, as the average distance between customers and their nearest depot decreases. In the following experiments, we investigate this trade-off to arrive at an estimation for *the value of opening more depots*.

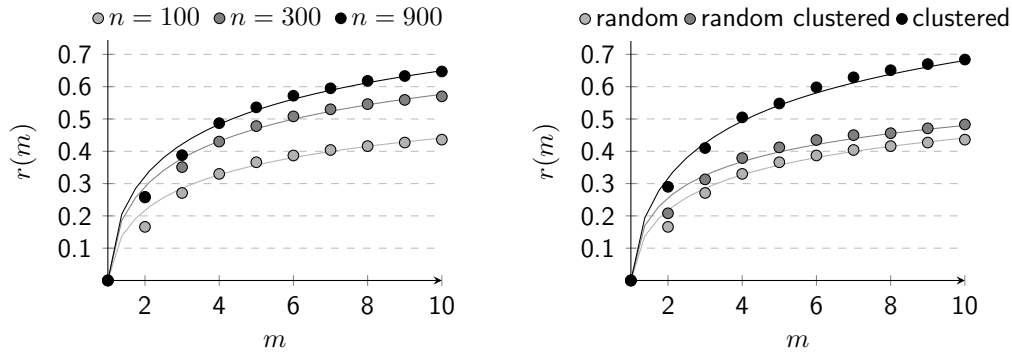
For each experiment, we sample 100 sets of MDVRP instances. Each set contains 10 instances with identical customer characteristics (location and demand), but different number and locations of the depots. The instances have an increasing number $m = \{1, 2, \ldots, 10\}$ of depots, and the location of the depots is computed with the k-means algorithm of the SciKit-learn library (Pedregosa et al. 2011). We use the k-means algorithm to open the available depots at good locations to answer the following research question as accurately as possible: Given n customer locations with certain characteristics, what is the reduction in routing costs when opening $m + 1$ depots in comparison to opening m depots, assuming that the depots can be opened at any location.

We then compute the routing costs for each instance with KGLS allowing $0.1 \cdot n$ seconds of runtime. Let R_m be the average routing costs of the 100 instances with m depots, then the target metric $r(m) = \frac{R_1 - R_m}{R_1}$ denotes the average cost reduction for the same instance with a single depot.

We perform four experiments and vary one of the relevant instance characteristics outlined by Uchoa et al. (2017) in each of them: (1) the number of customers $n \in \{100, 300, 900\}$, (2) the degree to which customers are clustered (random, random-clustered, clustered), (3) the variance in customer demand $d \in \{[1, 1], [1, 10], [1, 100]\}$ and (4) the average number of customers per route (the *route length*) $l \in \{3, 10, 30\}$. Clusters are generated in the same fashion as in Uchoa et al. (2017) where $k \in [3, 8]$ customers serve as cluster seeds in whose vicinity the other customers are placed with an exponentially decaying likelihood. The route length is varied by adjusting the vehicle capacity $Q = \lceil \frac{\sum_i d_i}{l} \rceil$.

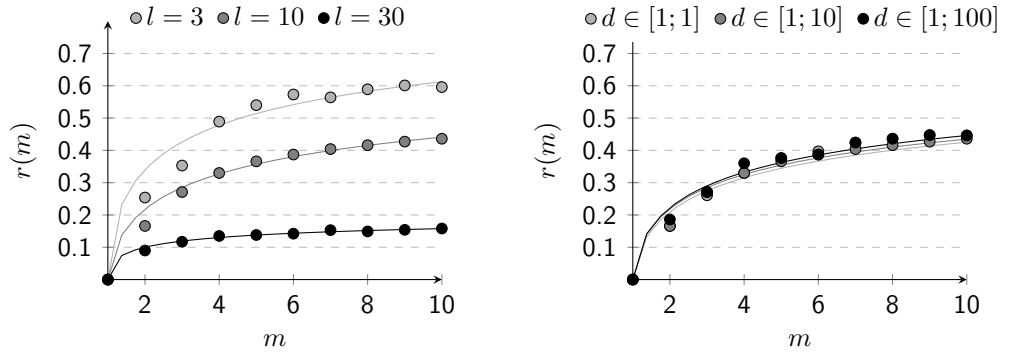
Fig. 1 plots the observed cost reductions $r(m)$ for each of the four experiments. The results confirm the hypothesis that more open depots generally result in larger cost reductions. However, the marginal benefit $r(m) - r(m-1)$ of having one more open depot decreases with growing m, resulting in a relationship that resembles the logarithmic function $r(m) = \alpha \cdot log(m)^{\beta}$. The parameters α and β are fitted with the software R (R Core Team 2014) for each experiment. For the baseline scenario $n = 100$, random customer distribution, $d \in [1, 10]$ and $l = 10$ we obtain $r_{base}(m) = 0.27 \cdot log(m)^{0.59}$.

Especially the number of customers n and the route length l appear to significantly determine the estimated savings $r(m)$. Given that both these parameters influence the



(a) Number of customers

(b) Customer clusters



(c) Number of customers per routes

(d) Variance in demand

Figure 1: Observed reduction in routing costs $r(m)$ when delivering from an increasing number of depots m compared to delivering from a single central depot for various instance properties. The plotted lines correspond to the fitted functions.

number of routes in a solution, we conclude that a larger number of routes (caused by a larger n or a smaller l) results in larger values for $r(m)$. To obtain an instance-specific estimate, we therefore include the minimal number of routes $t = \lceil \frac{\sum_i d_i}{Q} \rceil$ as a parameter in the fitted function with respect to our baseline $t_{base} = 10$:

$$r(m) = 0.27\sqrt{\frac{t}{10}} \cdot log(m)^{0.59}. \tag{1}$$

We observed that this factor accurately captures the savings for $t > 10$ and slightly overestimates the savings for $t < 10$.

Similarly, a higher degree of customer clustering appears to increase the benefit of more depots. However, the effect of clustering on savings is difficult to generalize. In the experiment, the k-means algorithm performs best (measured by the savings $r(m)$), if $m = k$ depots are placed as centroids in the k clusters and, thus, the number of clusters strongly impacts the savings curve $r(m)$. In LRP instances, however, it can generally not be assumed that candidate depots can be opened inside or close to customer clusters so that this effect vanishes. Therefore, we choose not to generalize the observed effect.

In contrast, the variance in demand has only a marginal effect.

Even though these functions have been derived based on empirical observations on a specific instance setup and thus present an approximation, rather than an analytically-proven relationship, they can be beneficial to estimate an upper bound for the number of open depots. The values $r(m)$ were derived under the assumption that the m depots are well-placed in a continuous space (with k-means). In an LRP instance with a limited number of given candidate depots, such good configurations might not be possible and $r(m)$ is likely to overestimate the possible cost reductions. Thus, $R_1 \cdot r(m)$ represents an estimate for the maximal cost reductions when opening m depots and $R_1 \cdot (r(m+1) - r(m))$ is an estimate for the maximal benefit of having a configuration with one additional open depot. If these benefits are outweighed by the opening cost of the additional depot, then it is not worthwhile to consider configurations with $m+1$ or more open depots. This idea is manifested in the estimated upper bound

$$m_C = \arg\min\{m \in \mathbb{N} \mid R_1 \cdot (r(m) - r(m-1)) < O_{\sigma(m)}\}, \tag{2}$$

where $\sigma(i)$ denotes the depot with the ith lowest opening costs and $O_{\sigma(m)}$ thus presents a lower bound for the opening costs of the most expensive depot in a depot configuration with m open depots.

In capacitated instances of the LRPs a minimal number of open depots might be required to serve all customers. Let $\sigma_W(i)$ denote the depot with the ith lowest capacity, then the upper bound for the minimal number of open depots required to serve all customers amounts to

$$m_W = \arg\min\{m \in \mathbb{N} \mid \sum_{j=1}^{m} W_{\sigma_W(j)} \geq \sum_{i \in N} d_i\}. \tag{3}$$

The estimated upper bound for capacitated instances is then derived as

$$U = \max(m_C, m_W). \tag{4}$$

All depot configurations with more open depots are eliminated from the search, reducing the number of candidates to $\sum_{i=1}^{U} \binom{m}{i}$.

5.2. Heuristic construction of depot configurations

If U is large or cannot be estimated because opening costs are negligible, the number of considered depot configurations need to be restricted even more. We propose a construction process which iteratively constructs depot configurations by combining promising depots in a spatially dispersed manner. In this process, the characteristics of individual depots are considered, and thus we shift the focus from the quantitative perspective above to a qualitative perspective, determining which depots should be opened with a higher priority. Individual depots should have (1) low opening costs, and (2) a proximity to customers. The depot configuration should be (3) in general distributed across the relevant area that is populated with customers, rather than being clustered in one spot (there might be exceptional cases).

We consider properties (1) and (2) by defining the *gross cost* G_j for each depot j. The gross costs are composed of the opening cost as well as the estimated routing costs from this depot. Let $L = \lceil \frac{Q}{\sum_{i \in N} d_i} \rceil$ denote the minimal number of routes in the routing solution, and assume that U depots are open, then each depot has $\lceil \frac{L}{U} \rceil$ outgoing routes on average. Under the assumption that the depots are likely to be connected with spatially close customers, the sum of the cost to the $2\lceil \frac{L}{U} \rceil$ closest customers thus present a rough approximation of the costs of incident edges. Let $\sigma_j(i)$ denote the i-th nearest customers to depot j, then the gross costs are defined by

$$G_j = \frac{\sum_{i=1}^{2\lceil \frac{L}{U} \rceil} c_{\sigma_j(i)j} + O_j}{W_j}. \tag{5}$$

The costs are weighted by capacity W_j to express the costs per available capacity unit.

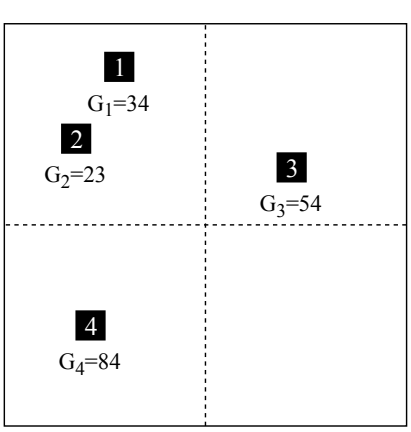
For property (3), the spatial dependencies between open depots have to be considered. We propose a divide and conquer approach in which the considered plane is iteratively split into smaller areas, for which depot sub-configurations are determined. For each $r \in \{1, 2, \ldots, \min(U, 16)\}$ the plane is split in a grid-like pattern into r isometric rectangular areas. The entire plane is divided into $rows = \lceil \frac{r}{cols} \rceil$ equidistant rows (where $cols = \lceil \sqrt{r} \rceil$), and each row is split into either $cols$ or $cols - 1$ areas, such that the number of all areas sum up to r.

In each area, the depots with the lowest gross costs are selected for sub-configurations: The first sub-configuration is the empty set, the next one contains the depot with the lowest gross costs, the next one contains the depot with the lowest and the second lowest gross costs, and so forth. A depot configuration is then constructed as the combination of sub-configurations from the r areas. Each such combination constitutes one candidate depot configuration. An example for $r = 4$ is given in Fig. 2. With increasing r, the areas become more granular and the number of sub-configuration combinations grows exponentially. Thus, the construction process is stopped prematurely once the upper limit of depot configurations M has been reached.

6. Computational experiments

In this section, we thoroughly analyse each component of PF. In particular, we experimentally compare different methods to arrive at an upper bound, investigate the impact of the heuristic construction method, and perform a detailed study on the number and granularity of filters. We arrive at an efficient PF configuration which is compared to several effective heuristics in the literature, both for the LRP, as for the variants and very-large-scale instances discussed in Section 4.4.

PF has been implemented in Java and all experiments are performed on an AMD Ryzen 3 1300X CPU working at 3.5GHz on Windows 10, using a single thread. It is available as an executable Java package at `http://antor.uantwerpen.be/LRP`



$$S_1 = \{\{\}, \{2\}, \{2,1\}\}$$
$$S_2 = \{\{\}, \{3\}\}$$
$$S_3 = \{\{\}, \{4\}\}$$
$$S_4 = \{\{\}\}$$

$$D_1 = \{3\}$$
$$D_2 = \{3,4\}$$
$$D_3 = \{4\}$$
$$D_4 = \{2\}$$
$$D_5 = \{2,3\}$$
$$D_6 = \{2,3,4\}$$
$$D_7 = \{2,4\}$$
$$D_8 = \{2,1\}$$
$$D_9 = \{2,1,3\}$$
$$D_{10} = \{2,1,3,4\}$$
$$D_{11} = \{2,1,4\}$$

Figure 2: Promising depot configurations are constructed by iteratively dividing the plane into an increasing number r of isometric regions (in this case $r = 4$). First, the gross costs G_i for each candidate depot i are computed, and a set of depot sub-configurations S_j is built for each region j by iteratively adding the depot with the next lowest gross costs to the previous configurations, starting with the empty set. Depot configurations D_k are then constructed by combining one sub-configuration from each region.

6.1. Sensitivity analysis

The crucial parameters for PF are the number and the granularity of the utilized filters. While more filters are expected to sift through the configurations more thoroughly and thereby increase the likelihood that the best configuration(s) survive, the overall computational time increases with each additional filter. Likewise, the granularity of a filter determines how many configurations are passed to the next filter, so that a higher granularity increases the likelihood that the best configuration(s) is passed on while more computational effort is required to eliminate the remaining configurations.

In the following experiments we investigate this trade-off to identify a computationally efficient filtering setup. We systematically test different combinations of the filters $(KGLS, 100)$, $(KGLS, 30)$, $(KGLS, 10)$ and $(KGLS, 3)$, denoted 1) – 15). Each combination obtains a time budget of $\frac{60 \cdot n}{100}$ seconds, which is equally divided among the involved filters. Thus, with more filters there is less time available for each of them. The best found depot configuration C_{best} is evaluated with KGLS for $\frac{30 \cdot n}{100}$ seconds, since in Arnold and Sörensen (2019) we found that KGLS requires about 60 seconds to almost optimally solve MDVRPs with $n = 200$. For filters with many input configurations, very short computation times are allocated to each configuration and, thus, minor deviations in actual processing time between two configurations within the same filter can already significantly impact the computed result. Thus, we map the allocated time to a fixed number of executed iterations in KGLS. As the basis for this mapping, we use the fact that KGLS performs about 100 iterations per second for $n = 100$, while scaling approximately linear in the problem size. Therefore, in a filter with 50 input configurations and a time budget of 30 seconds, each configuration is run for $\frac{30}{50} \cdot 100 = 60$ iterations. As a result, PF becomes a deterministic heuristic.

Likewise, we experimentally decided to set $M = 20,000$ to avoid that the first stage of PF (evaluation of each promising configuration with RCW) takes significantly more time than each of the subsequent stages.

Additionally, we validate the effectiveness of the initial reduction of configurations with the upper bound of open depots U and the subsequent heuristic construction. In alternative setups, U is replaced with previous methods from the literature. As suggested by Prins et al. (2006a) we compute the number of depots that are the nearest depot for at least one customer (denoted as NC). As second method, related to Chan and Baker (2005), we compute the minimal spanning tree out of all customer nodes and count the number of depots for which the distance to at least one customer is shorter than that customer's distances in the minimal tree (denoted as MST).

A sensitivity analysis is performed by testing the setups with $U - 1$ and $U + 1$, respectively. The effectiveness of the heuristic construction is tested by applying it on all possible configurations (denoted as $ELIM$). The necessity of plane division is tested by a setup in which the depots with the highest gross costs are opened, without any successive divisions (denoted as GRO). These last two alternatives are compared to the baseline in which $M = 20,000$ configurations are selected at random (denoted as RAN).

All setups are tested on the diverse instance set by Tuzun and Burke (1999). The 36 instances have 100–200 customers which are either clustered or randomly distributed, and 10–20 candidate depots, resulting in up to 1,048,575 possible depot configurations. The average gap to the best-known solutions over all instances is taken as the performance metric.

The results of this analysis are visualized in Fig. 3. We observe that most filter setups yield a similar performance. This finding suggests that PF performance is relatively insensitive with respect to the specific filter setup. The best results are achieved with either two or three filters which suggests that there is a fine balance between choosing too few and too many filtering stages. It is noticeable that the best setups have a mixture of fine-grained and coarse filters. The reason for this phenomenon can be explained by looking at setups 1) – 4). The more granular the filter (i.e., a small b), the larger is the bias towards configurations which show a fast convergence rate during evaluation. As a consequence, configurations that would have yielded excellent solutions (but only after some computational effort) are potentially removed too early, as seen in setup 4). It thus seems beneficial to start with more coarse filters which keep some configurations that do not yet yield excellent solutions (given the limited runtime of the respective stage), but might do so after a more accurate evaluation in subsequent stages. Overall, many setups perform similarly well, and we decided to use setup 13) with $P = \{(KGLS, 100), (KGLS, 10), (KGLS, 3)\}$ for all remaining experiments in this work since it performs slightly better than most others. Very-large-scale instances present an exception for which no filters $P = \{\}$ are used to limit runtime.

With $U - 1$ as upper bound the performance drops significantly and, thus, we can conclude that some of the best depot configurations have U open depots. On the other hand, $U + 1$ increases runtime without improving performance, which means that most,

if not all, good depot configurations have at most U open depots. The use of NC and MST increases the upper bounds further, and computational effort increases by almost 10 folds without achieving any performance gains. The heuristic construction procedure (without an upper bound) appears to find reasonably good configurations by itself and find solutions that are almost 6% better than if simply the depots with the highest gross costs are opened and 2% better than creating the same number of random configurations. This highlights the need for a procedure that diversifies the generation of configurations (in our case with an iterative plane division).

Instances with $m = 20$ require more time in the initial elimination stage, however, the particular good performance on those instances highlights the ability of PF to effectively filter among a vast number of possible configurations. The performance on instances with $n = 200$ is slightly worse than on instances with less customers, while the performance on clustered instances is almost identical to that on non-clustered instances.

6.2. Performance analysis

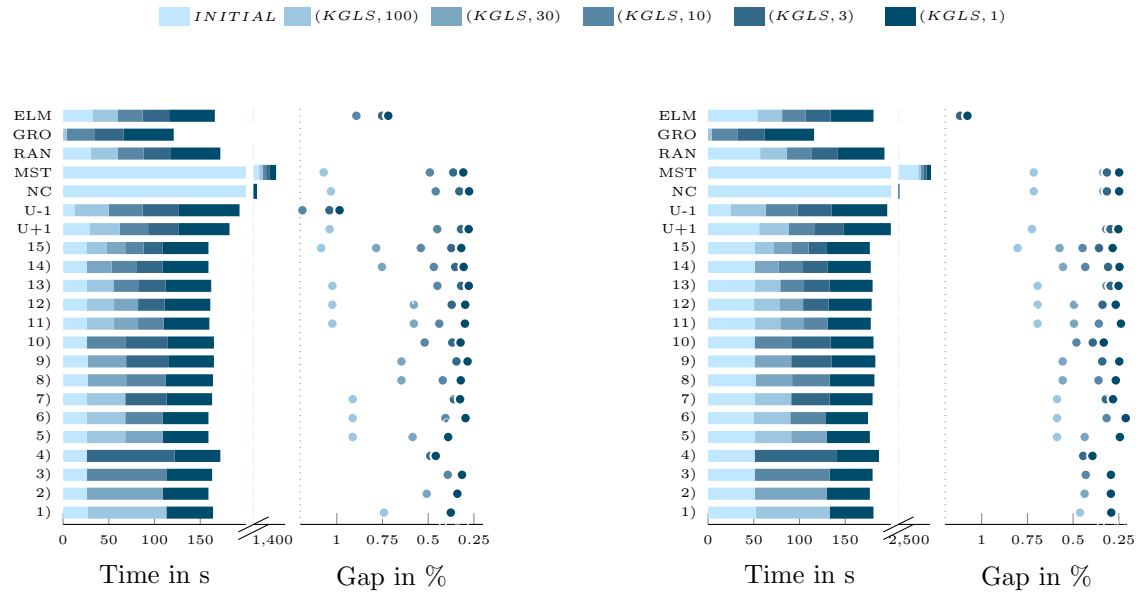
We start by discussing the instances used (Section 6.2.1), then discuss the benchmark algorithms in the literature (Section 6.2.2), and finally, compare these to the PF heuristic (Section 6.2.3). We restrict ourselves to a broad overview of the results of our heuristic, and refer to Tables 5 to 9 in Appendix for more detailed results. To demonstrate the effect of multi-threading, all experiments for the capacitated instances of the LRP have been run singe-threaded (PF_1) and on three threads (PF_3) on three physical cores on the same machine.

6.2.1. Instances

The PF heuristic is tested on a wide range of instance sets from the literature. Its performance on capacitated and uncapacitated instances is tested on three extensively used benchmark sets by Tuzun and Burke (1999), Prins et al. (2006b) and Barreto et al. (2007). These sets involve small and moderately-sized benchmark instances, some with clusters of customers and depots. These instances are complemented with the recently introduced set by Schneider and Löffler (2017), a rich and diverse instance set with a plethora of different instance characteristics and sizes. The scaling of PF is tested on the very-large-scale instances provided by Harks et al. (2013), containing instances with a large number of uniformly distributed customers and depots. The performance on 2E-LRPs is analyzed by using the instances by Nguyen et al. (2012), which are an extension of the $\mathbb{P}$-instances with an additional platform at a corner. Finally, the set by Villegas et al. (2010) is used to investigate PF's application on STTRPSD instances. A brief summary of all instance sets is provided in Table 1, and more details are given in Appendix.

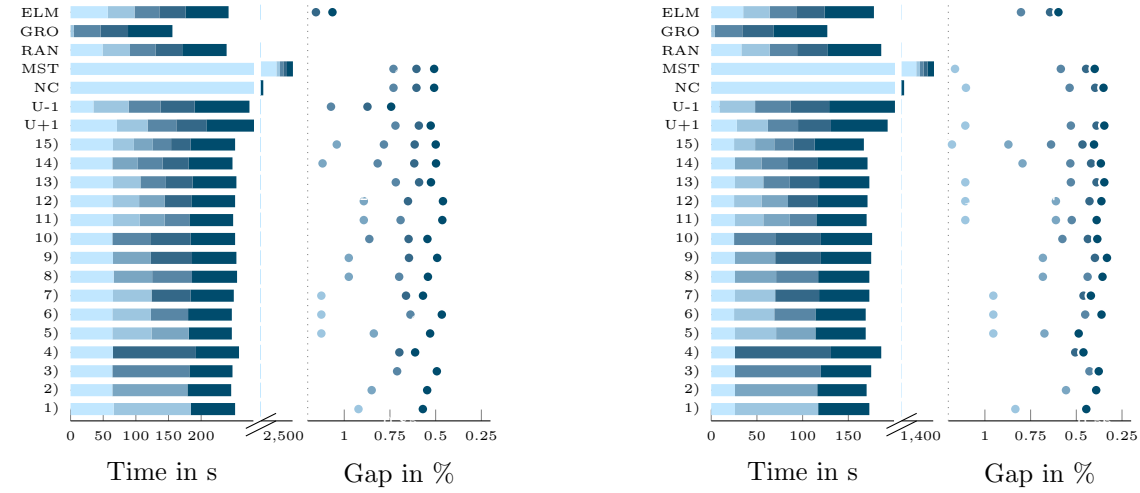
6.2.2. Benchmarks

The performance of PF is compared to the performance of the most effective heuristics for each problem type. For the LRP sets, Schneider and Drexl (2017) find that



(a) All instances

(b) Instances with $m = 20$



(c) Instances with $n = 200$

(d) Instances with customer clusters

Figure 3: Sensitivity analysis of different setups for PF. The colored bars indicate the average computational time of the respective stage within the given setup while the dots indicate the average gap to the best-known solutions. *INITIAL* represents the initial stage in which depot configurations are eliminated by means of the upper bound or heuristic construction.

Table 1: Overview of the different instance sets

Set	Reference	Type	Size	n	m	capacitated
$\mathbb{T}$	Tuzun and Burke (1999)	LRP	36	100-200	5-10	No
$\mathbb{P}$	Prins et al. (2006b)	LRP	30	20-200	5-10	Yes
$\mathbb{B}$	Barreto et al. (2007)	LRP	13	21-150	5-14	Partly
$\mathbb{S}$	Schneider and Löffler (2017)	LRP	220	100-600	5-30	Yes
$\mathbb{H}$	Harks et al. (2013)	LRP	27	1000-10,000	100-1000	No
$\mathbb{P}2\mathbb{E}$	Nguyen et al. (2012)	2E-LRP	30	20-200	5-10	Yes
$\mathbb{N}2\mathbb{E}$	Nguyen et al. (2012)	2E-LRP	24	25-200	5-10	Yes
$\mathbb{VTT}$	Villegas et al. (2010)	STTRPSD	32	25-200	5-20	No

Table 2: Test details about the benchmark heuristics.

Heuristic	Reference	Type	Stochastic	Runs	CPU Score
GRASP+ILP	Contardo et al. (2014)	LRP	Yes	10	1219
GVTNS	Escobar et al. (2014)	LRP	No	1	776
TBSA	Schneider and Löffler (2017)	LRP	Yes	5	1652
Approx+TSP	Harks et al. (2013)	LRP (large)	No	1	592
2-SH	Guemri et al. (2016)	LRP (large)	Yes	5	1248
LNS-2e	Breunig et al. (2016)	2E-LRP	Yes	10	1597
VNS-2e	Schwengerer et al. (2012)	2E-LRP	Yes	20	1132
ALNS-2e	Contardo et al. (2012)	2E-LRP	Yes	20	1244
MS-ELS	Villegas et al. (2010)	STTRPSD	Yes	10	801
MS-ILS	Villegas et al. (2010)	STTRPSD	Yes	10	801

these are the GRASP+ILP heuristic by Contardo et al. (2014), the granular variable neighborhood search (GVTNS) by Escobar et al. (2014), and the tree-based search algorithm (TBSA) by Schneider and Löffler (2017). TBSA outperforms all other LRP heuristics on the classical benchmark sets, and we also compare it with PF on the more recent $\mathbb{S}$-instances. The $\mathbb{T}$-instances have only been tackled by Harks et al. (2013) (Approx+TSP) and the two-stage heuristic by Guemri et al. (2016) (2-SH). We use the performance of both methods as a benchmark. On the 2E-LRP-instances the large neighbourhood search by Breunig et al. (2016) (LNS-2e), the variable neighbourhood search by Schwengerer et al. (2012) (VNS-2e), and the adaptive large neighborhood search by Contardo et al. (2012) (ALNS-2e) are the most effective heuristics. The $\mathbb{VTT}$-set was successfully solved with the multi-start evolutionary local search (MS-ELS), and the multi-start iterated local search (MS-ILS) by Villegas et al. (2010). An overview about these heuristics is given in Table 2.

For each heuristic with stochastic components we report the average performance over all executed runs, since the average appears to be a fairer metric than the best observed values over several runs (Birattari and Dorigo 2007), especially when comparing stochastic and deterministic algorithms.

To allow a fairer comparison of computation times, all CPU times are normalized as suggested in Schneider and Drexl (2017). The reported times are multiplied by the single thread speed score of the used CPU (PassMark Software 2018) and divided by the score of an i7-4790 processor running at 3.60GHz (a score of 2290). We want to remark that the CPU is not the only component that determines the execution speed of a system, it nonetheless represents the best available measure to normalize times across systems.

6.2.3. Results on LRP instances

Table 3 summarizes the performance of PF on LRP benchmark sets in comparison to the most effective heuristics in literature. Detailed results per instance can be found in Tables 5 to 9 in Appendix.

On the smaller LRP benchmark sets, PF computes high-quality solutions close to the best-known solutions for almost all instances. On average, PF achieves a better accuracy than GVTNS and GRASP+ILP and a similar performance than the current state-of-the-art heuristic TBSA on all three benchmark sets. All of these solutions are computed in short computation times. Multi-threading can further improve performance, three physical cores reduces computational time by about 43%. Note that solutions are for some instances sightly better when using three cores, since three configurations are evaluated final stage (compared to one when using a single thread).

On the largest and diverse $\mathbb{S}$-set, PF outperforms the state-of-the-art LRP heuristic in the literature. This is especially true on larger instances with $n \geq 400$ customers, on which PF performs significantly better. The shorter computation times can be attributed to a good scaling of computation time as a function of the instance size, which is largely due to the drastic elimination of depot configurations. The improvements in terms of solution quality can be tied to the significant improvement of PF on many

Table 3: Benchmark results of different heuristics for LRP instance sets. The gap is reported as the average difference to the BKS in % across all executed runs, and the times express the average execution time of a single run in seconds.

LRP	GRASP+ILP		GVTNS		TBSA_{speed}		$\text{TBSA}_{quality}$		PF_1		PF_3	
	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time
$\mathbb{T}(36)$	0.66	1379	0.86	68	0.57	48	0.15	725	0.34	132	0.34	75
$\mathbb{P}(30)$	0.38	619	0.43	31	0.20	23	0.02	452	0.14	85	0.14	49
$\mathbb{B}(13)$	0.59	141	0.63	135	0.08	10	0.05	171	0.04	71	0.04	56
	0.54	886	0.66	46	0.35	32	0.08	522	0.21	104	0.21	62
$\mathbb{S}(102)$	$n \leq 300$						0.16	845	0.02	265	0.00	137
(100)	$n > 300$						0.39	4700	-0.34	779	-0.35	411
							0.27	1979	-0.15	519	-0.17	272
All(281)							**0.22**	**2121**	**-0.05**	**401**	**-0.06**	**213**

Large-scale LRP	Approx+TSP		2-SH		PF_1	
	Gap	Time	Gap	Time	Gap	Time
$\mathbb{H}(27)$	4.89	57	3.68	36	-6.38	187

instances of subtype 'e' as shown in Table 6 and Table 7 in Appendix. These instances have the characteristic that "[...] a large number of interesting configurations with both a lower number of high-capacity depots and a higher number of low-capacity depots exists." (Schneider and Löffler 2017). A good performance on these instances types thus requires the consideration of a wide range of diverse depot configurations. While this task might pose a hindrance to local search based approaches (since promising configurations might be structurally far apart), and ILP-based approaches (since a large number of configurations has to be considered), PF appears to consistently detect good configurations. In total PF improved the BKSs of 63 instances, and since PF is completely deterministic, these BKS are computed in every run.

PF can also tackle very-large-scale problems successfully, as demonstrated by the results on the $\mathbb{H}$-instances. It significantly improves the best-known results on almost all instances in slightly larger computation times. On instances of which the optimal solution only has a few open depots, PF does not perform as well, see e.g., row $M2-3$ in Table 9 in Appendix. On those instances it is preferable to open depots in the center of the created regions during heuristic construction of configurations, rather than choosing depots with low gross costs.

6.2.4. Results on LRP variants

As summarized in Table 4, PF outperforms previous heuristics on 2E-LRP instances, computing on average better solutions in shorter times. On both the $\mathbb{P}2\mathbb{E}$-instances and

Table 4: Benchmark results of different heuristics for 2E-LRPs and STTRPSDs. The gap is reported as the average difference to the BKS in % across all executed runs, and the times express the average execution time of a single run in seconds.

2E-LRP	LNS-2e		VNS-2e		ALNS-2e		PF	
$\mathbb{P}2\mathbb{E}$	1.13	385	0.49	150	0.82	253	0.11	69
$\mathbb{N}2\mathbb{E}$	1.15	335	0.90	136	0.48	104	0.20	59
	1.14	307	0.67	144	0.67	187	0.15	64
STTRPSD			MS-ELS		MS-ILS		PF	
$\mathbb{VTT}$			0.31	113	0.25	139	0.27	106

the $\mathbb{N}2\mathbb{E}$-instances almost all but three computed solutions (on which the gap is larger than 0.5%) correspond to the BKS or have a very small gap. These results indicate that the transformation of a 2E-LRP into a combination of LRP and VRP constitutes a promising simplification for the considered instances.

STTRPSD instances can also be solved effectively with PF, obtaining a similar performance than a dedicated heuristic. This performance highlights the ability of PF to cope with situations in which no upper bound for the number of open depots can be estimated, and thus a huge number of configurations has to be evaluated. This observation validates the effectiveness of the heuristic construction of depot configurations.

In summary, PF is a generic heuristic framework that can be successfully applied to various problem variants of the LRP. Combined with an effective routing heuristic, it is competitive with the most effective heuristics in the literature on traditional instance sets and STTRPSD instances, while for larger LRP instances, very-large-scale LRP instances, and 2E-LRP instances it outperforms existing heuristics. This performance can also be seen as an empirical validation of the estimated upper bound and the effectiveness of the subsequent filtering stages. It also confirms that high-quality depot configurations are quickly identified as such.

7. Conclusions and future work

In this paper, we have designed a solution approach for the location–routing problem (LRP) and variants thereof. Our framework uses an efficient heuristic to solve the routing subproblem, in combination with a progressive filtering strategy to remove unpromising depot configurations as early as possible. At each iteration of the filtering stage, the accuracy of the computed routing solutions increases at the expense of an increase in computation time. An upper bound is estimated on the number of open depots to reduce the number of depot configurations that need to be evaluated. We observe experimentally that a finer filtering with more stages can improve the performance, while sufficient time should be granted to each filter. The same heuristic, with some minor modifications, can also be used to solve very-large-scale LRPs, 2E-LRPs

and STTRPSDs.

Computational tests on many benchmark sets show that the resulting heuristic can effectively solve a wide range of instances within short computation times and either matches or improves upon the performances of the most effective heuristics in the literature. The framework performs especially well on instances of large or very large size, and instances with a wide range of good depot configurations, on which previous solutions are improved by several percent.

The heuristic design is flexible and scalable, and therefore useful in practical cases where implementation time is limited. A straightforward Clarke-Wright implementation is sufficient to obtain satisfactory results in seconds, while the use of the effective routing heuristic KGLS renders our approach competitive with the best results in the literature. This algorithmic design allows developers and researchers to re-use their routing implementations for the tested problem variants, and possibly even more. Two conditions imposed by such a framework are that feasible MDVRP solutions can be computed quickly, and that the number of initial configurations can be reduced successfully, for instance by defining an upper bound. Even though we observed that the second condition can be circumvented in the case that no upper bound can be estimated, the enumeration of the entire space of configurations should be avoided, and more research in this direction appears promising.

References

M. Albareda-Sambola and J. Rodríguez-Pereira. Location-routing and location-arc routing. In G. Laporte, S. Nickel, and F. Saldanha da Gama, editors, *Location Science*, pages 431–451. Springer, 2019.

F. Arnold and K. Sörensen. Knowledge-guided local search for the vehicle routing problem. *Computers & Operations Research*, 105:32 – 46, 2019. ISSN 0305-0548.

F. Arnold, M. Gendreau, and K. Sörensen. Efficiently solving very large-scale routing problems. *Computers & Operations Research*, 107:32 – 42, 2019. ISSN 0305-0548.

S. Barreto, C. Ferreira, J. Paixao, and B. Santos Sousa. Using clustering analysis in a capacitated location-routing problem. *European Journal of Operational Research*, 179(3): 968–977, 2007.

M. Birattari and M. Dorigo. How to assess and report the performance of a stochastic algorithm on a benchmark problem: mean or best result on a number of runs? *Optimization Letters*, 1(3):309–311, 2007.

U. Breunig, V. Schmid, R.F. Hartl, and Th. Vidal. A large neighbourhood based heuristic for two-echelon routing problems. *Computers & Operations Research*, 76:208–225, 2016.

Y. Chan and S.F. Baker. The multiple depot, multiple traveling salesmen facility-location problem: Vehicle range, service frequency, and heuristic implementations. *Mathematical and Computer Modelling*, 41(8-9):1035–1053, 2005.

G.U. Clarke and J.W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, 1964.

C. Contardo, V. Hemmelmayr, and T.G. Crainic. Lower and upper bounds for the two-echelon capacitated location-routing problem. *Computers & Operations Research*, 39 (12):3185–3199, 2012.

C. Contardo, J.-F. Cordeau, and B. Gendron. A GRASP+ILP-based metaheuristic for the capacitated location-routing problem. *Journal of Heuristics*, 20(1):1–38, 2014.

A. Del Pia and C. Filippi. A variable neighborhood descent algorithm for a real waste collection problem with mobile depots. *International Transactions in Operational Research*, 13(2): 125–141, 2006.

M. Drexl and M. Schneider. A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research*, 241(2):283 – 308, 2015. ISSN 0377-2217.

J.W. Escobar and P. Linfati, R.and Toth. A two-phase hybrid heuristic algorithm for the capacitated location-routing problem. *Computers & Operations Research*, 40(1):70–79, 2013.

J.W. Escobar, R. Linfati, M.G. Baldoquin, and P. Toth. A granular variable tabu neighborhood search for the capacitated location-routing problem. *Transportation Research Part B: Methodological*, 67:344–356, 2014.

O. Guemri, B. Beldjilali, A. Bekrar, and G. Belalem. Two-stage heuristic algorithm for the large-scale capacitated location routing problem. *International Journal of Mathematical Modelling and Numerical Optimisation*, 7(1):97–119, 2016.

T. Harks, F.G. König, and J. Matuschke. Approximation algorithms for capacitated location routing. *Transportation Science*, 47(1):3–22, 2013.

S. Lin and B.W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):498–516, 1973.

R.B. Lopes, C. Ferreira, B. Sousa Santos, and S. Barreto. A taxonomical analysis, current methods and objectives on location-routing problems. *International Transactions in Operational Research*, 20(6):795–822, 2013.

V.P. Nguyen, C. Prins, and C. Prodhon. A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem. *Engineering Applications of Artificial Intelligence*, 25(1):56–71, 2012.

PassMark Software. CPU benchmarks. `https://www.cpubenchmark.net/`, 2018. Accessed: 2018-02-05.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

C. Prins, C. Prodhon, and R. Wolfler Calvo. A memetic algorithm with population management (ma— pm) for the capacitated location-routing problem. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 183–194. Springer, 2006a.

C. Prins, C. Prodhon, and R. Wolfler Calvo. Solving the capacitated location-routing problem by a grasp complemented by a learning process and a path relinking. *4OR: A Quarterly Journal of Operations Research*, 4(3):221–238, 2006b.

C. Prodhon and C. Prins. A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1):1–17, 2014.

R Core Team. R: A language and environment for statistical computing, 2014. URL `http://www.R-project.org/`.

S. Salhi and G.K. Rand. The effect of ignoring routes when locating depots. *European Journal of Operational Research*, 39(2):150–156, 1989.

P. Schittekat and K. Sörensen. OR practice—supporting 3PL decisions in the automotive industry by generating diverse solutions to a large-scale location-routing problem. *Operations Research*, 57(5):1058–1067, 2009.

M. Schneider and M. Drexl. A survey of the standard location-routing problem. *Annals of Operations Research*, 259(1-2):389–414, 2017.

M. Schneider and M. Löffler. Large composite neighborhoods for the capacitated location-routing problem. *Transportation Science*, 53(1):1–18, 2017.

M. Schwengerer, S. Pirkwieser, and G.R. Raidl. A variable neighborhood search approach for the two-echelon location-routing problem. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 13–24. Springer, 2012.

C.-J. Ting and C.-H. Chen. A multiple ant colony optimization algorithm for the capacitated location routing problem. *International Journal of Production Economics*, 141(1):34–44, 2013.

D. Tuzun and L.I. Burke. A two-phase tabu search approach to the location routing problem. *European Journal of Operational Research*, 116(1):87–99, 1999.

E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, Th. Vidal, and A. Subramanian. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858, 2017.

Th. Vidal, G. Laporte, and P. Matl. A concise guide to existing and emerging vehicle routing problem variants. *European Journal of Operational Research*, 286(2):401–416, 2020.

J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, and N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5):780 – 794, 2010.

C. Voudouris and E.P.K. Tsang. *Guided local search*. Springer, 2003.

Appendix: Detailed results of performance analysis

The following tables contain the detailed results of the performance analysis on the LRP instances by Tuzun and Burke (1999), Prins et al. (2006b), Barreto et al. (2007) and Schneider and Löffler (2017), the large-scale instances by Harks et al. (2013), the 2E-LRP instances by Nguyen et al. (2012) and the STTRPSD instances by Villegas et al. (2010). The values for the best-known solutions (BKS) for the LRP instances are taken from Schneider and Drexl (2017), from Guemri et al. (2016) for the $\mathbb{H}$-set, from Breunig et al. (2016) for the $\mathbb{P}2\mathbb{E}$- and $\mathbb{N}2\mathbb{E}$-set, and from Villegas et al. (2010) for the $\mathbb{VTT}$-set.

The distances in all instances are calculated with a Euclidean norm. The distances for the $\mathbb{P}$, $\mathbb{S}$, $\mathbb{P}2\mathbb{E}$, $\mathbb{N}2\mathbb{E}$, and $\mathbb{VTT}$-set are multiplied by 100 and rounded up to the next integer. On the 2E-LRP instances the distance costs on the first level are multiplied by two. The $\mathbb{T}$, $\mathbb{P}$, $\mathbb{B}$, $\mathbb{P}2\mathbb{E}$, $\mathbb{N}2\mathbb{E}$ and $\mathbb{VTT}$-set are available at `http://prodhonc.free.fr/Instances`, the $\mathbb{S}$-set is attached to the paper of Schneider and Löffler (2017), and the $\mathbb{H}$-set can be found at `http://www.coga.tu-berlin.de/clrlib`.

During testing we found two inconsistencies. On some $\mathbb{H}$-instances some values for d_i exceed Q, and to obtain feasible solutions we set $d_i = Q$ in these cases. Instance 200-10-3b of the $\mathbb{P}2\mathbb{E}$-set does not include a fixed costs for routes on the first level as discussed by Breunig et al. (2016). We set the fixed cost to 5000 as done by the aforementioned authors and do not include the instance in the computation of the average gap to avoid inconsistencies with other results.

Each instance is solved once with PF since its behavior is deterministic. We report the objective value of this single run together with the computational time, the estimated upper bound U and number of open depots m_{best} in the best found solution.

Table 5: Results on the classical instance sets $\mathbb{T}$, $\mathbb{P}$ and $\mathbb{B}$. Gap to the BKS in %, time in seconds, estimated upper bounds of open depots U and the number of open depots m_{best} in the best found solution.

Instance ($\mathbb{P}$ and $\mathbb{B}$)	BKS	PF					Instance ($\mathbb{T}$)	BKS	PF				
		Value	Gap	Time	m_{best}	U			Value	Gap	Time	m_{best}	U
20-5-1	54793	54793	0.00	20	3	3	P111112	1467.68	1468.29	0.04	92	3	4
20-5-1b	39104	39104	0.00	26	2	2	P111122	1448.37	1448.37	0.00	95	2	4
20-5-2	48908	48908	0.00	16	3	4	P111212	1394.80	1394.80	0.00	88	2	3
20-5-2b	37542	37542	0.00	26	2	3	P111222	1432.29	1432.29	0.00	83	2	4
50-5-1	90111	90111	0.00	43	3	3	P112112	1167.16	1167.16	0.00	117	2	3
50-5-1b	63242	63242	0.00	62	2	3	P112122	1102.24	1102.24	0.00	119	2	3
50-5-2	88293	88298	0.01	46	3	3	P112212	791.66	791.66	0.00	103	2	3
50-5-2b	67308	67308	0.00	58	3	3	P112222	728.30	728.30	0.00	105	2	3
50-5-2bBIS	51822	51822	0.00	56	3	3	P113112	1238.24	1238.67	0.03	105	3	3
50-5-2BIS	84055	84055	0.00	70	3	3	P113122	1245.30	1245.31	0.00	93	3	3
50-5-3	86203	86203	0.00	49	2	3	P113212	902.26	902.34	0.01	110	3	3
50-5-3b	61830	61830	0.00	62	2	3	P113222	1018.29	1018.29	0.00	99	3	3
200-10-1	474850	474937	0.02	228	3	4	P121112	2237.73	2245.07	0.33	213	3	5
200-10-1b	375177	376485	0.35	225	3	4	P121122	2137.45	2174.72	1.74	206	4	5
200-10-2	448077	448850	0.17	223	3	4	P121212	2195.17	2204.52	0.43	193	4	6
200-10-2b	373696	374266	0.15	217	3	4	P121222	2214.86	2224.02	0.41	211	4	6
200-10-3	469433	470978	0.33	261	3	4	P122112	2070.43	2074.28	0.19	312	3	5
200-10-3b	362320	362841	0.14	269	3	4	P122122	1685.52	1693.53	0.47	309	3	5
100-10-1	287661	288395	0.26	138	3	4	P122212	1449.93	1449.91	0.00	284	2	4
100-10-1b	230989	232342	0.59	109	3	4	P122222	1082.46	1083.16	0.06	203	3	5
100-10-2	243590	243590	0.00	89	3	4	P123112	1942.23	1961.49	0.99	215	4	5
100-10-2b	203988	203988	0.00	91	3	4	P123122	1910.08	1921.03	0.57	220	4	5
100-10-3	250882	252918	0.81	122	3	4	P123212	1761.11	1800.08	2.21	242	3	5
100-10-3b	203114	204567	0.72	109	3	4	P123222	1390.86	1391.99	0.08	184	5	5
100-5-1	274814	275505	0.25	106	3	3	P131112	1892.17	1897.92	0.30	124	3	5
100-5-1b	213568	213971	0.19	95	3	3	P131122	1819.68	1820.32	0.04	126	4	4
100-5-2	193671	193671	0.00	92	2	3	P131212	1960.02	1960.02	0.00	132	3	4
100-5-2b	157095	157110	0.01	82	2	3	P131222	1792.77	1792.77	0.00	120	3	5
100-5-3	200079	200237	0.08	73	2	3	P132112	1443.32	1448.81	0.38	172	2	4
100-5-3b	152441	152441	0.00	75	2	3	P132122	1429.30	1444.77	1.08	209	2	4
							P132212	1204.42	1204.64	0.02	166	3	4
50x5	565.6	565.6	0.00	52	2	3	P132222	924.68	931.55	0.74	140	3	3
75x10	848.9	848.9	0.00	69	3	4	P133112	1694.18	1709.34	0.89	157	3	4
100x10	833.4	836.7	0.39	107	2	4	P133122	1392.01	1400.50	0.61	159	3	4
Daskin95-88x8	355.8	355.8	0.00	158	2	2	P133212	1197.95	1200.14	0.18	121	3	4
Daskin95-150x10	43919.9	43919.9	0.00	274	3	3	P133222	1151.37	1156.61	0.46	164	3	4
Gaskell67-21x5	424.9	424.9	0.00	19	2	3							
Gaskell67-22x5	585.1	585.1	0.00	40	1	3							
Gaskell67-29x5	512.1	512.1	0.00	41	2	3							
Gaskell67-32x5	562.2	562.2	0.00	55	1	3							
Gaskell67-32x5	504.3	504.3	0.01	40	1	3							
Gaskell67-36x5	460.4	460.4	0.00	51	1	3							
Min92-27x5	3062	3062.0	0.00	40	2	3							
Min92-134x8	5709	5713.0	0.07	175	3	5							

Table 6: Results on the LRP instance set $\mathbb{S}$. Gap to the BKS in %. time in seconds. estimated upper bounds of open depots U and the number of open depots m_{best} in the best found solution.

Instance	BKS	PF					Instance	BKS	PF				
		Value	Gap	Time	m_{best}	U			Value	Gap	Time	m_{best}	U
100-5-1c	134,516	134,687	0.13	112	5	5	200-15-2e	712,524	737081	3.45	1242	8	12
100-5-1d	275,749	276,154	0.15	122	3	3	200-15-3a	455,676	456081	0.09	258	3	4
100-5-1e	292,311	292,565	0.09	159	2	5	200-15-3b	357,086	357233	0.04	249	3	4
100-5-2c	83,989	85,051	1.26	81	4	5	200-15-3c	141,129	141703	0.41	223	12	15
100-5-2d	242,266	242,739	0.20	101	3	3	200-15-3d	877,638	878358	0.08	242	8	8
100-5-2e	253,888	254,085	0.08	169	1	5	200-15-3e	816,377	816579	0.02	295	9	13
100-5-3c	87,555	87,555	0.00	66	5	5	200-15-4a	433,268	434165	0.21	362	3	4
100-5-3d	226,783	226,920	0.06	74	3	3	200-15-4b	349,269	350509	0.36	339	3	4
100-5-3e	252,603	252,677	0.03	119	1	5	200-15-4c	143,772	144536	0.53	256	10	15
100-5-4a	255,853	255,869	0.01	142	3	4	200-15-4d	828,144	828711	0.07	296	8	8
100-5-4b	214,425	214,531	0.05	107	3	4	200-15-4e	700,202	701825	0.23	394	5	13
100-5-4c	98,129	98,199	0.07	137	3	5	300-15-1a	856,267	856306	0.00	379	3	4
100-5-4d	250,315	251,380	0.43	131	3	3	300-15-1b	622,412	623644	0.20	399	3	4
100-5-4e	211,159	211,444	0.13	205	4	4	300-15-1c	366,770	366675	-0.03	455	15	15
100-10-1c	92,629	92,979	0.38	77	10	10	300-15-1d	1,339,010	1341270	0.17	381	8	8
100-10-1d	363,930	363,930	0.00	102	5	5	300-15-1e	1,217,690	1219197	0.12	553	6	13
100-10-1e	344,322	344,897	0.17	115	4	9	300-15-2a	759,999	762089	0.28	367	3	4
100-10-2c	84,717	84,817	0.12	86	8	10	300-15-2b	557,912	557525	-0.07	366	3	4
100-10-2d	343,252	343,252	0.00	97	5	5	300-15-2c	311,558	312374	0.26	457	10	15
100-10-2e	332,900	333,778	0.26	123	4	9	300-15-2d	1,301,863	1305933	0.31	352	8	8
100-10-3c	85,618	85,369	-0.29	79	10	10	300-15-2e	1,272,700	1276145	0.27	575	9	13
100-10-3d	329,990	329,990	0.00	118	5	5	300-15-3a	778,023	778590	0.07	354	3	4
100-10-3e	318,156	318,226	0.02	137	4	9	300-15-3b	594,073	593892	-0.03	348	3	4
100-10-4a	253,892	253,471	-0.17	142	3	4	300-15-3c	341,712	342206	0.14	360	12	15
100-10-4b	211,354	211,361	0.00	125	3	4	300-15-3d	1,358,223	1355955	-0.17	359	8	8
100-10-4c	86,215	87,277	1.23	104	9	10	300-15-3e	1,286,877	1289607	0.21	604	8	13
100-10-4d	328,251	328,420	0.05	131	5	5	300-15-4a	747,730	750135	0.32	439	3	4
100-10-4e	308,866	310,134	0.41	169	4	9	300-15-4b	559,877	560352	0.08	472	3	4
200-10-1c	156,087	157,428	0.86	199	10	10	300-15-4c	304,254	303984	-0.09	543	14	15
200-10-1d	638,452	638,372	-0.01	249	5	5	300-15-4d	1,288,091	1289757	0.13	458	8	8
200-10-1e	599,463	600,954	0.25	330	4	9	300-15-4e	1,173,516	1174438	0.08	621	5	13
200-10-2c	144,337	144,666	0.23	185	8	10	300-20-1a	1,009,840	945545	-6.37	513	4	5
200-10-2d	663,814	664,234	0.06	220	5	5	300-20-1b	739,604	740915	0.18	514	4	5
200-10-2e	619,037	619,262	0.04	249	4	9	300-20-1c	364,096	364303	0.06	514	20	20
200-10-3c	184,885	186,112	0.66	247	8	10	300-20-1d	1,575,390	1579766	0.28	401	10	10
200-10-3d	640,357	641,424	0.17	248	5	5	300-20-1e	1,391,567	1320924	-5.08	575	5	18
200-10-3e	604,617	606,919	0.38	252	4	9	300-20-2a	909,306	909376	0.01	553	4	5
200-10-4a	452,870	453,435	0.12	297	3	4	300-20-2b	695,524	695155	-0.05	558	4	5
200-10-4b	369,951	369,821	-0.04	271	3	4	300-20-2c	299,425	309529	3.37	470	14	20
200-10-4c	144,407	144,940	0.37	231	10	10	300-20-2d	1,569,139	1571166	0.13	369	10	10
200-10-4d	618,590	618,795	0.03	279	5	5	300-20-2e	1,386,386	1284831	-7.33	655	3	18
200-10-4e	562,854	564,383	0.27	318	4	9	300-20-3a	929,901	930992	0.12	469	4	5
200-15-1a	461,203	462,359	0.25	236	3	4	300-20-3b	751,307	750844	-0.06	489	4	5
200-15-1b	367,397	367,330	-0.02	232	3	4	300-20-3c	305,771	307684	0.63	488	18	20
200-15-1c	148,218	150,091	1.26	215	14	15	300-20-3d	1,539,008	1541385	0.15	373	10	10
200-15-1d	813,941	814,072	0.02	248	8	8	300-20-3e	1,289,734	1265041	-1.91	649	7	18
200-15-1e	708,837	709,259	0.06	326	6	13	300-20-4a	859,474	859446	0.00	574	4	5
200-15-2a	513,893	514,199	0.06	265	3	4	300-20-4b	687,930	688604	0.10	559	4	5
200-15-2b	406,843	407,449	0.15	265	3	4	300-20-4c	300,285	301424	0.38	642	17	20
200-15-2c	135,051	135,633	0.43	230	14	15	300-20-4d	1,540,194	1542127	0.13	653	10	10
200-15-2d	811,722	813,280	0.19	249	8	8	300-20-4e	1,344,056	1330488	-1.01	805	7	18
										0.02	322		

Table 7: Results on the LRP instance set $\mathbb{S}$. Gap to the BKS in %. time in seconds. estimated upper bounds of open depots U and the number of open depots m_{best} in the best found solution.

Instance	BKS	PF					Instance	BKS	PF				
		Value	Gap	Time	m_{best}	U			Value	Gap	Time	m_{best}	U
400-20-1a	1,140,605	1,140,975	0.03	733	4	5	500-25-3a	1,725,918	1756884	1.79	723	4	5
400-20-1b	880,393	876,157	-0.48	725	4	5	500-25-3b	1,305,521	1314903	0.72	693	4	5
400-20-1c	467,755	471,031	0.70	813	18	20	500-25-3c	581,425	580688	-0.13	817	20	25
400-20-1d	1,956,824	1,957,929	0.06	805	10	10	500-25-3d	3,248,557	3249805	0.04	760	13	13
400-20-1e	1,748,962	1,645,475	-5.92	701	7	18	500-25-3e	2,768,174	2697267	-2.56	1056	7	22
400-20-2a	1,053,445	1,055,570	0.20	671	4	5	500-25-4a	1,655,514	1655310	-0.01	714	4	5
400-20-2b	829,494	828,932	-0.07	636	4	5	500-25-4b	1,263,496	1260960	-0.20	720	4	5
400-20-2c	394,712	395,668	0.24	647	17	20	500-25-4c	664,089	669088	0.75	1229	22	25
400-20-2d	1,875,072	1,878,113	0.16	546	10	10	500-25-4d	3,362,588	3370668	0.24	967	13	13
400-20-2e	1,608,600	1,562,339	-2.88	776	7	18	500-25-4e	2,630,867	2647733	0.64	1164	7	22
400-20-3a	1,098,989	1,100,714	0.16	828	4	5	500-30-1a	1,984,150	1999892	0.79	924	5	6
400-20-3b	849,555	847,618	-0.23	640	4	5	500-30-1b	1,538,027	1537821	-0.01	881	5	6
400-20-3c	391,928	393,648	0.44	647	17	20	500-30-1c	614,433	630143	2.56	841	29	30
400-20-3d	1,929,284	1,936,927	0.40	517	10	10	500-30-1d	3,742,922	3762156	0.51	1061	15	15
400-20-3e	1,778,315	1,679,271	-5.57	714	7	18	500-30-1e	3,485,608	3246339	-6.86	986	8	26
400-20-4a	1,081,452	1,083,252	0.17	706	4	5	500-30-2a	1,820,346	1820556	0.01	899	5	7
400-20-4b	842,078	843,238	0.14	717	4	5	500-30-2b	1,452,171	1452028	-0.01	971	5	7
400-20-4c	351,715	357,250	1.57	776	17	20	500-30-2c	649,471	650913	0.22	1009	20	30
400-20-4d	1,834,809	1,845,315	0.57	544	10	10	500-30-2d	3,815,306	3819650	0.11	1234	15	15
400-20-4e	1,620,575	1,558,411	-3.84	818	7	18	500-30-2e	3,293,153	3249618	-1.32	1086	8	26
400-25-1a	1,156,187	1,156,802	0.05	507	4	5	500-30-3a	1,782,554	1788808	0.35	876	5	6
400-25-1b	890,566	889,828	-0.08	501	4	5	500-30-3b	1,422,148	1424533	0.17	802	5	6
400-25-1c	395,268	412,212	4.29	609	23	25	500-30-3c	570,866	571418	0.10	1036	23	30
400-25-1d	2,341,499	2,350,173	0.37	650	13	13	500-30-3d	3,690,995	3710414	0.53	1382	15	15
400-25-1e	2,053,366	1,890,676	-7.92	730	7	22	500-30-3e	3,171,977	3059470	-3.55	1111	8	26
400-25-2a	1,091,595	1,102,122	0.96	490	4	5	500-30-4a	1,716,476	1725178	0.51	1214	5	7
400-25-2b	869,254	875,593	0.73	508	4	5	500-30-4b	1,398,401	1402136	0.27	1155	5	7
400-25-2c	360,923	362,684	0.49	644	19	25	500-30-4c	562,731	562542	-0.03	1071	23	30
400-25-2d	2,351,903	2,356,224	0.18	699	13	13	500-30-4d	3,708,479	3710414	0.05	1382	15	15
400-25-2e	1,954,300	1,910,645	-2.23	870	9	22	500-30-4e	3,194,234	3059470	-4.22	1111	8	26
400-25-3a	1,105,783	1,122,031	1.47	454	4	5	600-30-1a	2,198,674	2208578	0.45	1381	5	6
400-25-3b	862,180	871,206	1.05	477	4	5	600-30-1b	1,691,805	1697630	0.34	1346	5	6
400-25-3c	393,783	399,436	1.44	706	20	25	600-30-1c	748,714	746485	-0.30	1140	30	30
400-25-3d	2,321,358	2,340,658	0.83	527	13	13	600-30-1d	4,213,337	4214905	0.04	1564	15	15
400-25-3e	1,946,952	1,901,148	-2.35	878	9	22	600-30-1e	3,737,075	3570518	-4.46	1380	8	26
400-25-4a	1,015,654	1,016,670	0.10	546	4	5	600-30-2a	2,017,760	2023484	0.28	1105	5	6
400-25-4b	801,722	803,237	0.19	532	4	5	600-30-2b	1,602,833	1601675	-0.07	1019	5	6
400-25-4c	380,824	382,491	0.44	736	21	25	600-30-2c	634,787	646907	1.91	1368	21	30
400-25-4d	2,362,571	2,367,103	0.19	719	13	13	600-30-2d	4,163,772	4201095	0.90	1493	15	15
400-25-4e	1,992,633	1,942,603	-2.51	913	9	22	600-30-2e	3,682,117	3572959	-2.96	1672	8	26
500-25-1a	1,773,409	1,798,898	1.44	705	4	5	600-30-3a	2,082,824	2103960	1.01	1168	5	6
500-25-1b	1,331,827	1,349,058	1.29	709	4	5	600-30-3b	1,615,623	1623113	0.46	1247	5	6
500-25-1c	673,495	671,756	-0.26	1193	25	25	600-30-3c	662,569	683582	3.17	1311	24	30
500-25-1d	3,325,312	3,322,248	-0.09	833	13	13	600-30-3d	4,068,474	4126907	1.44	1724	15	15
500-25-1e	2,971,616	2,692,197	-9.40	1322	7	22	600-30-3e	3,496,852	3488544	-0.24	1526	11	26
500-25-2a	1,619,689	1,620,927	0.08	713	4	5	600-30-4a	1,940,218	1942190	0.10	1325	5	6
500-25-2b	1,252,748	1,251,667	-0.09	681	4	5	600-30-4b	1,560,237	1555080	-0.33	1366	5	6
500-25-2c	574,794	576,080	0.22	856	20	25	600-30-4c	707,288	714196	0.98	1808	26	30
500-25-2d	3,338,585	3,345,184	0.20	834	13	13	600-30-4d	4,150,775	4164022	0.32	2391	15	15
500-25-2e	2,802,823	2,729,944	-2.60	1337	7	22	600-30-4e	3,622,885	3543712	-2.19	1890	8	26
										-0.34	949		

Table 8: Results on the 2-LRP instance sets $\mathbb{P}2\mathbb{E}$ and $\mathbb{N}2\mathbb{E}$. Gap to the BKS in %, time in seconds, estimated upper bounds of open depots U and the number of open depots m_{best} in the best found solution.

Instance ($\mathbb{P}2\mathbb{E}$)	BKS	PF					Instance ($\mathbb{N}2\mathbb{E}$)	BKS	PF				
		Value	Gap	Time	m_{best}	U			Value	Gap	Time	m_{best}	U
20-5-1	89075	89075	0.00	24	3	3	25-5N	80370	80370	0.00	31	2	3
20-5-1b	61863	61863	0.00	26	2	2	25-5Nb	64562	64562	0.00	55	1	3
20-5-2	84478	84478	0.00	20	3	4	25-5MN	78947	78947	0.00	31	1	3
20-5-2b	60838	60838	0.00	28	2	3	25-5MNb	64438	64438	0.00	42	1	3
50-5-1	130843	130843	0.00	46	2	3	50-5N	137815	137815	0.00	45	3	3
50-5-1b	101530	101530	0.00	58	2	3	50-5Nb	110094	110094	0.00	75	2	3
50-5-2	131825	131825	0.00	43	3	3	50-5MN	123484	123484	0.00	41	3	4
50-5-2b	110332	110332	0.00	51	3	3	50-5MNb	105401	105401	0.00	52	3	3
50-5-2BIS	122599	122599	0.00	63	3	3	50-10N	115725	115725	0.00	55	3	3
50-5-2bBIS	105696	107310	1.53	69	3	3	50-10Nb	87315	87315	0.00	75	2	3
50-5-3	128379	128379	0.00	45	2	3	50-10MN	135519	135519	0.00	41	3	4
50-5-3b	104006	104006	0.00	54	2	3	50-10MNb	110613	110613	0.00	61	3	3
100-5-1	318134	318842	0.22	103	3	3	100-5N	193228	194255	0.53	64	4	4
100-5-1b	256878	256888	0.00	95	3	3	100-5Nb	158927	158927	0.00	91	4	5
100-5-2	231305	231305	0.00	91	2	3	100-5MN	204682	204848	0.08	58	4	5
100-5-2b	194728	194778	0.03	81	2	3	100-5MNb	165744	166087	0.21	81	4	5
100-5-3	244071	244958	0.36	77	2	3	100-10N	209952	210538	0.28	79	5	5
100-5-3b	194110	194110	0.00	86	2	3	100-10Nb	155489	155489	0.00	101	4	5
100-10-1	351243	351468	0.06	121	3	4	100-10MN	201275	201275	0.00	73	4	5
100-10-1b	297167	297513	0.12	111	3	4	100-10MNb	170625	170625	0.00	81	4	4
100-10-2	304438	304438	0.00	95	3	4	200-10N	345267	343232	-0.59	165	8	8
100-10-2b	263873	263876	0.00	97	3	4	200-10Nb	256171	256403	0.09	206	7	7
100-10-3	310200	310148	-0.02	140	3	4	200-10MN	323801	330259	1.99	165	6	8
100-10-3b	260328	260671	0.13	120	3	4	200-10MNb	287076	293246	2.15	202	7	7
200-10-1	548703	549194	0.09	208	3	4							
200-10-1b	445301	445658	0.08	211	3	4							
200-10-2	497451	498530	0.22	191	3	4							
200-10-2b	422668	422987	0.08	185	3	4							
200-10-3	527162	529202	0.39	242	3	4							
200-10-3b	401672	416355	3.66	228	3	4							
			0.11	96						0.20	82		

Table 9: Results on the very-large-scale LRP instance set $\mathbb{H}$ and the STTRPSD instance set $\mathbb{VTT}$. Gap to the BKS in %, time in seconds, estimated upper bounds of open depots U and the number of open depots m_{best} in the best found solution.

Instance	BKS	PF					Instance	BKS	PF				
($\mathbb{H}$)		Value	Gap	Time	m_{best}	U	($\mathbb{VTT}$)		Value	Gap	Time	m_{best}	U
M 1-1	12014.2	11026.4	-8.22	33	40	100	25-5-1-c	405.46	405.46	0.00	40	4	5
M 1-2	3470.3	3157.3	-9.01	32	12	32	25-5-2-c	374.79	374.79	0.00	168	1	5
M 1-3	2478.9	2431.8	-1.86	31	6	18	25-5-1-rd	584.03	584.03	0.00	171	4	5
M 2-1	15760.5	14718.6	-6.61	33	28	85	25-5-2-rd	508.48	508.48	0.00	184	1	5
M 2-2	4468.7	4019.5	-10.04	31	6	11	25-10-1-c	386.45	386.45	0.00	89	4	10
M 2-3	2620.8	2701.1	3.10	31	2	4	25-10-2-c	380.86	380.86	0.00	93	1	10
M 3-1	19839.5	17986.3	-9.34	32	18	49	25-10-1-rd	573.96	573.96	0.00	44	4	10
M 3-2	5207.4	4673.3	-10.25	30	6	7	25-10-2-rd	506.37	506.48	0.02	66	1	10
M 3-3	2779.3	2854.8	2.73	31	1	3	50-5-1-c	583.07	583.07	0.00	52	5	5
L 1-1	29538.1	27813.0	-5.84	222	177	200	50-5-2-c	516.98	516.98	0.00	217	1	5
L 1-2	8106.1	7577.7	-6.52	192	55	162	50-5-1-rd	870.51	870.51	0.00	50	3	5
L 1-3	5463.7	5392.5	-1.29	167	12	60	50-5-2-rd	766.03	766.03	0.00	97	3	5
L 2-1	50229.7	40305.4	-19.76	275	96	200	50-10-1-c	387.83	389.07	0.32	59	7	10
L 2-2	11551.8	10751.2	-6.92	164	23	54	50-10-2-c	367.01	367.01	0.00	107	3	10
L 2-3	6624.5	6016.0	-9.18	160	4	11	50-10-1-rd	811.28	811.28	0.00	55	5	10
L 3-1	54976.6	51245.7	-6.79	326	55	200	50-10-2-rd	731.53	731.53	0.00	114	1	10
L 3-2	13426.2	12658.2	-5.72	161	14	30	100-10-1-c	614.02	614.02	0.00	71	6	10
L 3-3	6966.5	6423.9	-7.78	160	4	7	100-10-2-c	547.44	547.44	0.00	139	5	10
XL 1-1	43939.8	45009.0	2.44	859	193	200	100-10-1-rd	1275.76	1271.78	-0.31	86	6	10
XL 1-2	11867.3	10826.0	-8.77	613	108	200	100-10-2-rd	1097.28	1097.28	0.00	135	4	10
XL 1-3	7754.7	7632.6	-1.57	459	25	130	100-20-1-c	642.61	644.95	0.36	82	6	20
XL 2-1	68118.7	62947.9	-7.59	849	168	200	100-20-2-c	581.56	594.70	2.26	136	3	20
XL 2-2	17638.8	16300.2	-7.58	422	30	121	100-20-1-rd	1143.10	1147.60	0.39	95	12	20
XL 2-3	9296.1	8809.8	-5.23	342	8	20	100-20-2-rd	1060.75	1067.14	0.60	145	4	20
XL 3-1	85509.5	78600.4	-8.08	630	95	200	200-10-1-c	822.52	821.59	-0.11	150	9	10
XL 3-2	20659.7	18885.4	-8.59	354	22	65	200-10-2-c	714.33	712.29	-0.29	196	7	10
XL 3-3	10389.9	9543.6	-8.15	349	4	11	200-10-1-rd	1761.10	1759.07	-0.12	189	8	10
							200-10-2-rd	1445.94	1445.94	0.00	451	5	10
							200-20-1-c	909.46	925.76	1.79	253	6	20
							200-20-2-c	815.51	815.87	0.04	403	6	20
							200-20-1-rd	1614.18	1652.19	2.35	206	12	20
							200-20-2-rd	1413.32	1433.78	1.45	366	7	20
			-6.39	259						0.27	147		